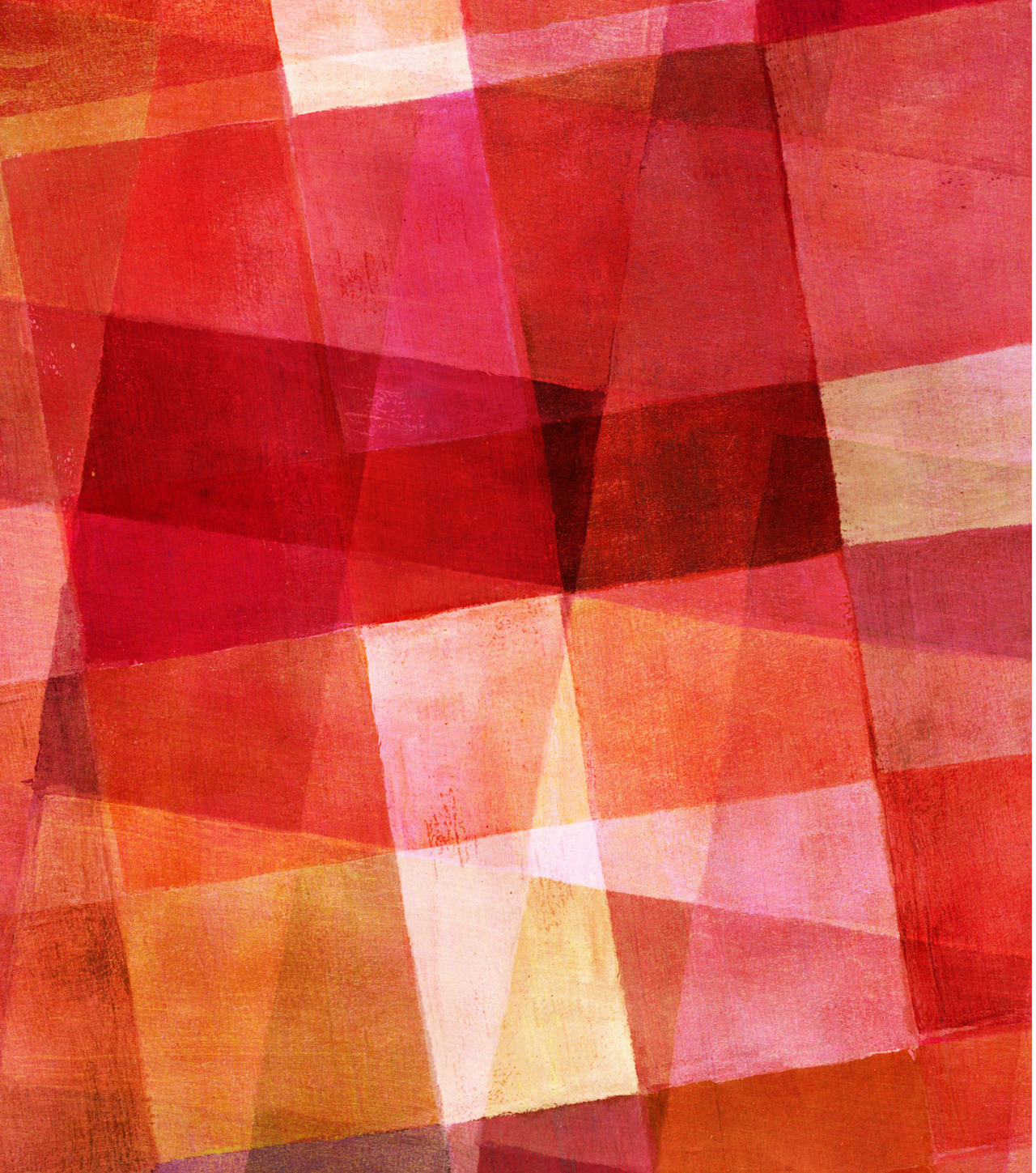
PROGRAMAÇÃO EM LINGUAGEM SQL

CINEL

Tecnologia e o Futuro num só Cent



CONVERT

SELECT + CONVERT

A função CONVERT tem um parâmetro opcional chamado style que pode ser usado para formatar uma data.

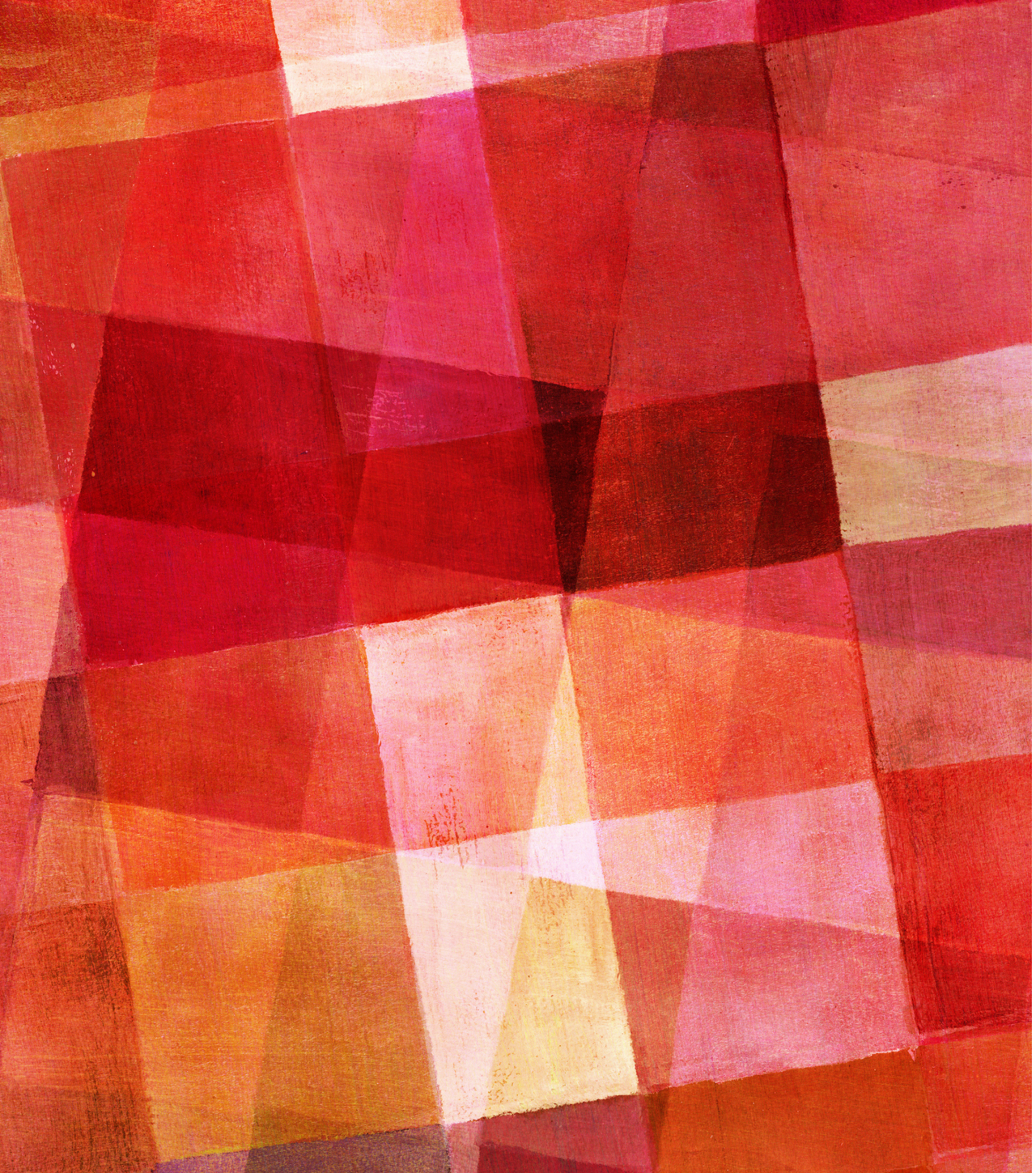
Com esta função é possível formar datas juntamente com a função DATEPART, bem como converter a data para um formato mais desejado

CONVERT(<data type, usually varchar>,<date>,<style>)

SELECT + CONVERT

```
--1 The hard way!

SELECT CAST(DATEPART(YYYY,GETDATE()) AS
VARCHAR) + '/' +

   CAST(DATEPART(MM,GETDATE()) AS VARCHAR) +

   '/' +  CAST(DATEPART(DD,GETDATE()) AS
VARCHAR) AS DateCast;

--2 The easy way!

SELECT CONVERT(VARCHAR,GETDATE(),111) AS
DateConvert;
```

```
--3

USE AdventureWorks2019

GO

SELECT CONVERT(VARCHAR,OrderDate,1) AS "1",

   CONVERT(VARCHAR,OrderDate,101) AS "101",

   CONVERT(VARCHAR,OrderDate,2) AS "2",

   CONVERT(VARCHAR,OrderDate,102) AS "102"

FROM Sales.SalesOrderHeader

WHERE SalesOrderID in (43659,43714,60621);
```



FORMAT

CINEL

Tecnologia e o Futuro num só Centr

SELECT + FORMAT

O objetivo principal é simplificar a conversão de valores de data / hora em valores de string.

Outro objetivo da função de formato é converter valores de data / hora em equivalências culturais.

FORMAT (value, format [, culture])

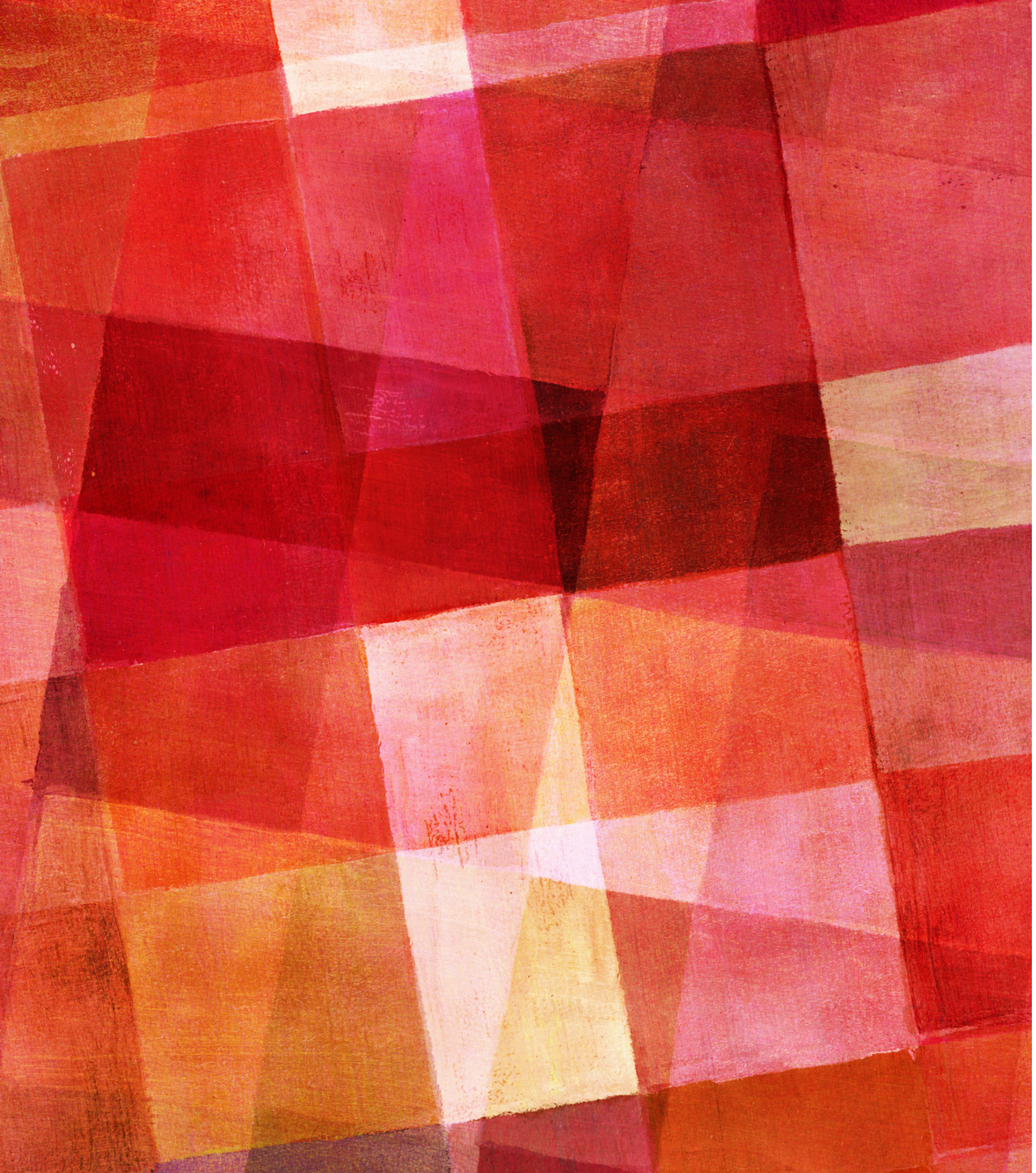
SELECT + FORMAT

```
USE AdventureWorks2019
GO
SELECT FORMAT( GETDATE(),'dd', 'en-US' ) AS Result),
SELECT FORMAT( GETDATE(), 'd/M/y', 'en-US' ) AS Result,
SELECT FORMAT( GETDATE(), 'dd/MM/yyyy', 'en-US' ) AS Result
FROM Sales.SalesOrderHeader
WHERE SalesOrderID in (43659,43714,60621);
```

SELECT + FORMAT

CINEL
Tecnologia e o Futuro num só Centr

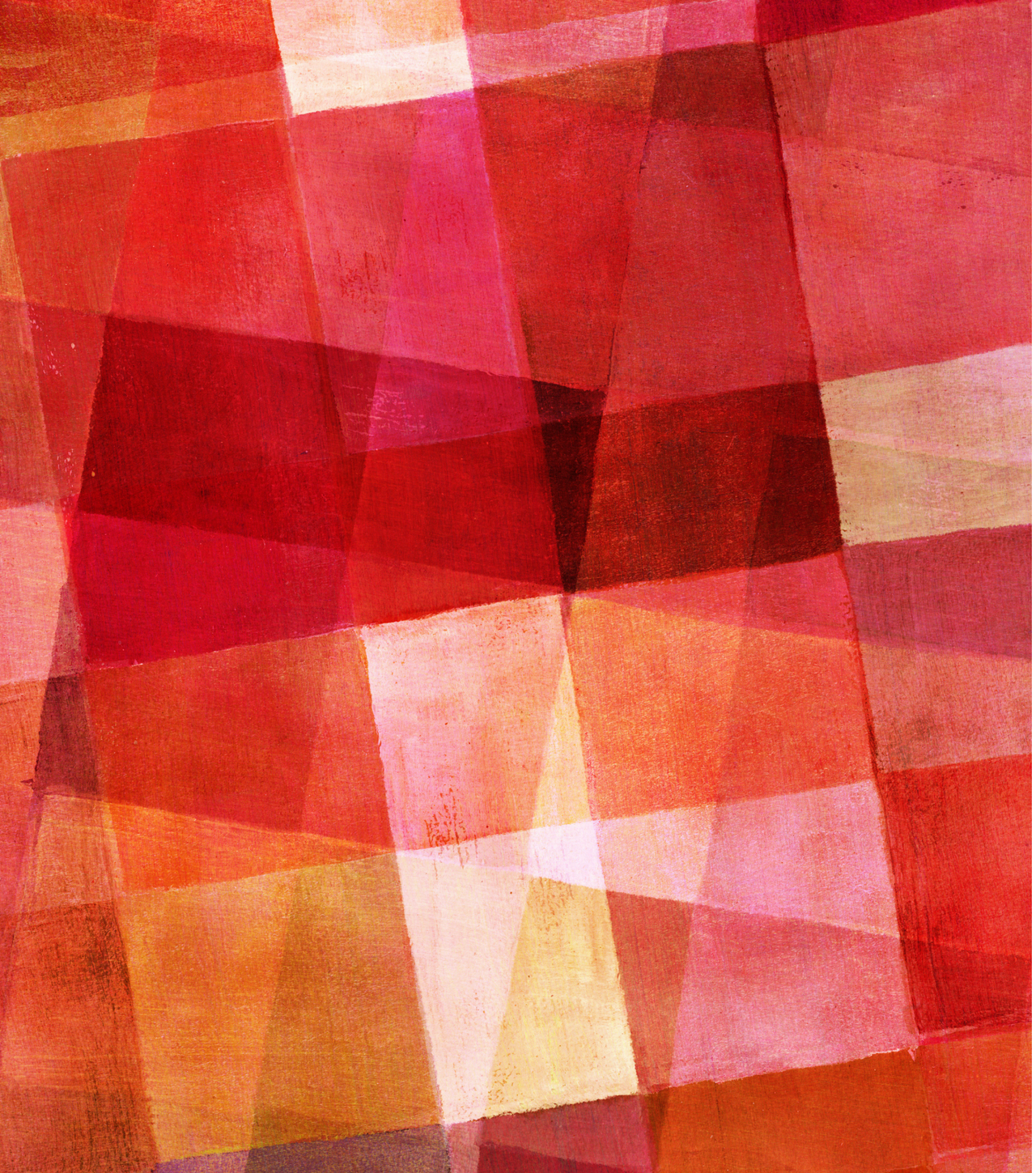
```
SELECT DATEFROMPARTS(2012, 3, 10) AS RESULT

SELECT TIMEFROMPARTS(12, 10, 32, 0, 0) AS RESULT

SELECT DATETIME2FROMPARTS (2012, 3, 10, 12, 10, 32, 0, 0) AS RESULT
```



CONCATENATE

SELECT + CONCATENATE


```
USE AdventureWorks2019;
GO


SELECT BusinessEntityID, FirstName + ' ' + MiddleName +
    ' ' + LastName AS "Full Name"
FROM Person.Person;
```



CONCAT

SELECT + CONCAT

A instrução CONCAT pega qualquer número de strings como argumentos e os concatena automaticamente.

Os valores podem ser passados para a instrução CONCAT como variáveis ou como strings regulares.

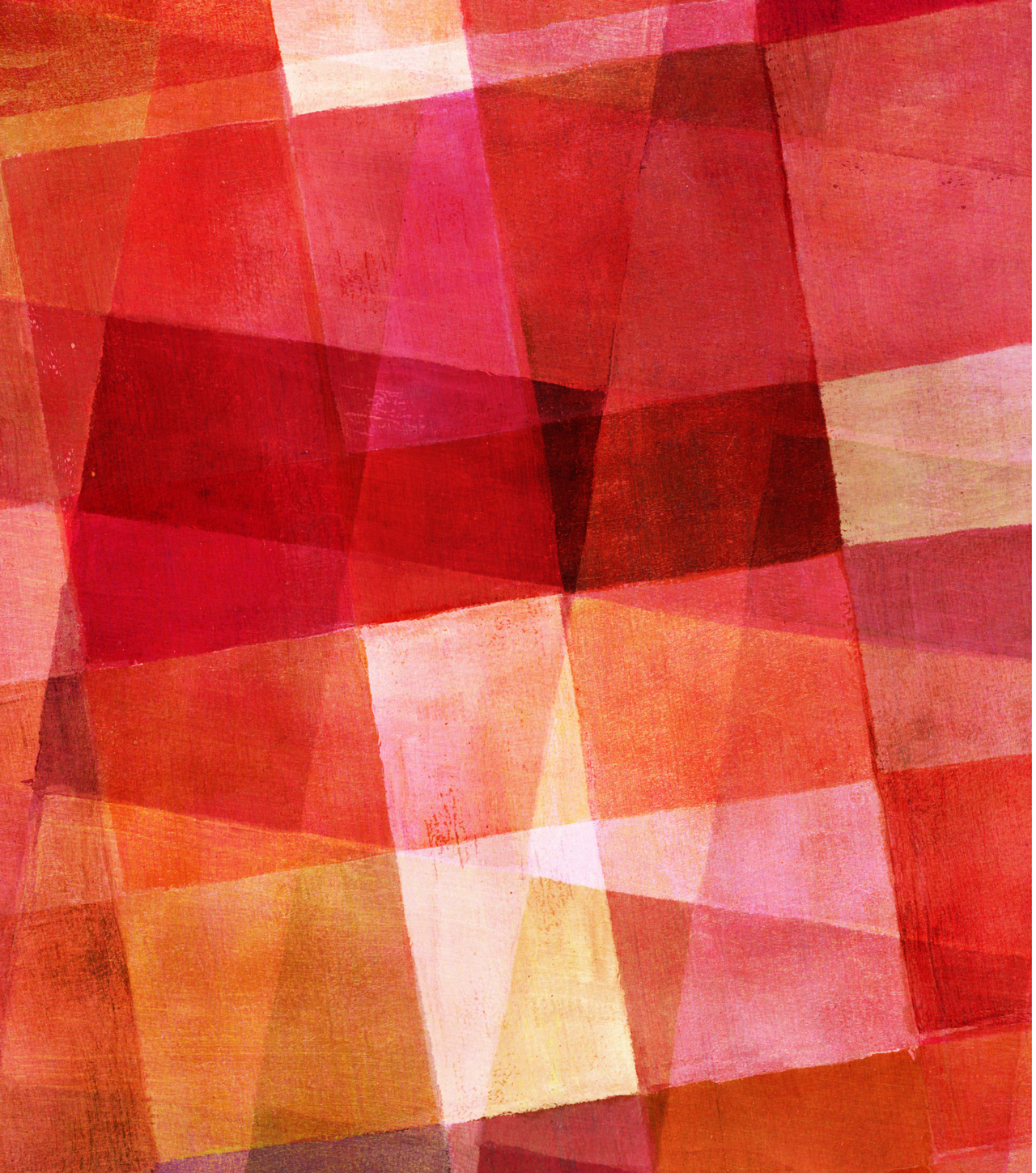
A saída é sempre convertida implicitamente em um tipo de dados de string

SELECT + CONCAT

```
-- Simple CONCAT statement

SELECT CONCAT ('I ', 'love', ' writing', ' T-SQL') AS RESULT;
```

```
--Using variable with CONCAT

DECLARE @a VARCHAR(30) = 'My birthday is on '

DECLARE @b DATE = '08/25/1980'

SELECT CONCAT (@a, @b) AS RESULT;
```

```
--Using CONCAT with table rows

USE AdventureWorks2012

SELECT CONCAT (AddressLine1, PostalCode) AS Address

FROM Person.Address;
```



ISNULL/COALESCE

SELECT + ISNULL/COALESCE

Duas funções estão disponíveis para substituir valores NULL por outro valor.

A primeira função, ISNULL, requer dois parâmetros: o valor a ser verificado e a substituição por valores NULL.

O COALESCE funciona de maneira um pouco diferente. COALESCE tomará qualquer número de parâmetros e retornará o primeiro valor diferente de NULL.

SELECT + ISNULL/COALESCE

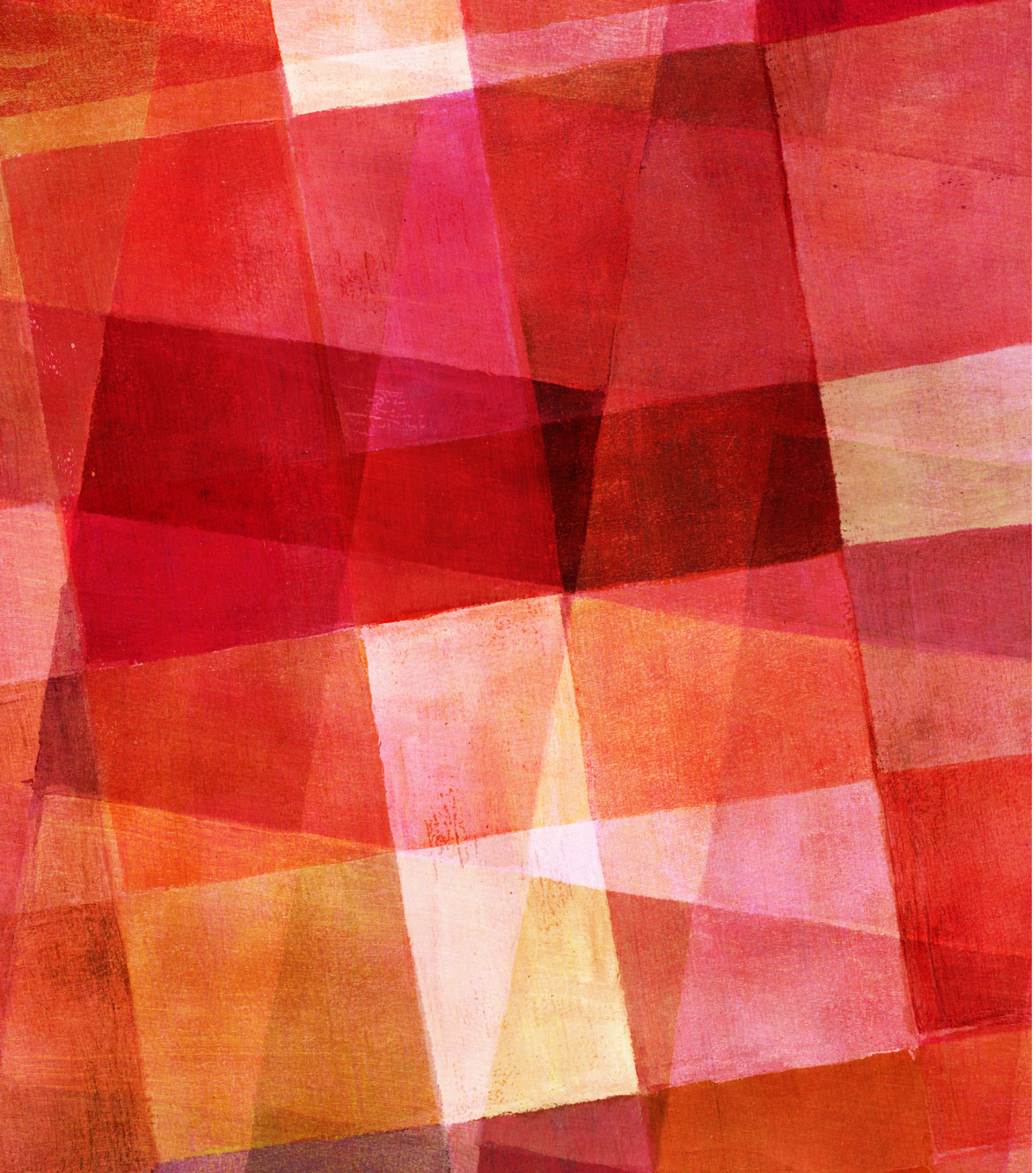
```
USE AdventureWorks2012;

GO

--1

SELECT BusinessEntityID, FirstName + ' ' +
ISNULL(MiddleName,'') +

    ' ' + LastName AS "Full Name"

FROM Person.Person;

--2

SELECT BusinessEntityID, FirstName + ISNULL(' '
+ MiddleName,'') +

    ' ' + LastName AS "Full Name"

FROM Person.Person;
```

```
--3

SELECT BusinessEntityID,
FirstName + COALESCE(' ' +
MiddleName,'') +

    ' ' + LastName AS "Full Name"

FROM Person.Person
```

SELECT + ISNULL/COALESCE

```
USE AdventureWorks2012
GO
--1
SELECT CAST(BusinessEntityID AS NVARCHAR) + ': ' +
LastName
    + ', ' + FirstName AS ID_Name
FROM Person.Person;
--2
SELECT CONVERT(NVARCHAR(10),BusinessEntityID) + ': ' +
LastName
    + ', ' + FirstName AS ID_Name
FROM Person.Person;
```

```
--3
SELECT BusinessEntityID,
BusinessEntityID + 1 AS "Adds 1",
    CAST(BusinessEntityID AS
NVARCHAR(10)) + '1'AS "Appends 1"
FROM Person.Person;
```



ABS

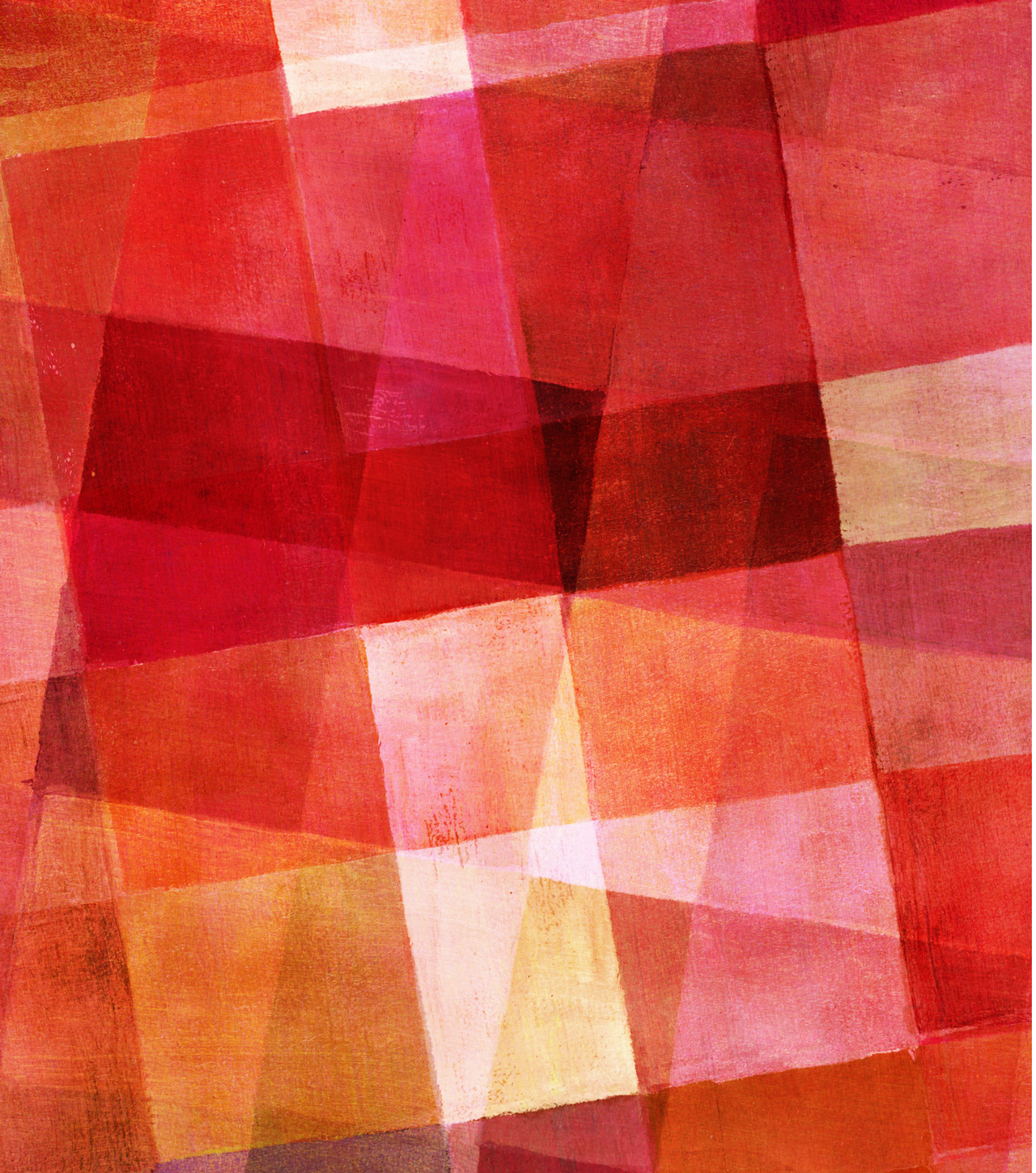
CINEL

Tecnologia e o Futuro num só Centr

SELECT + ABS

A função ABS retorna o valor absoluto do número - a diferença entre o número e zero.

```
SELECT ABS(2) AS "2", ABS(-2) AS "-2"
```



POWER

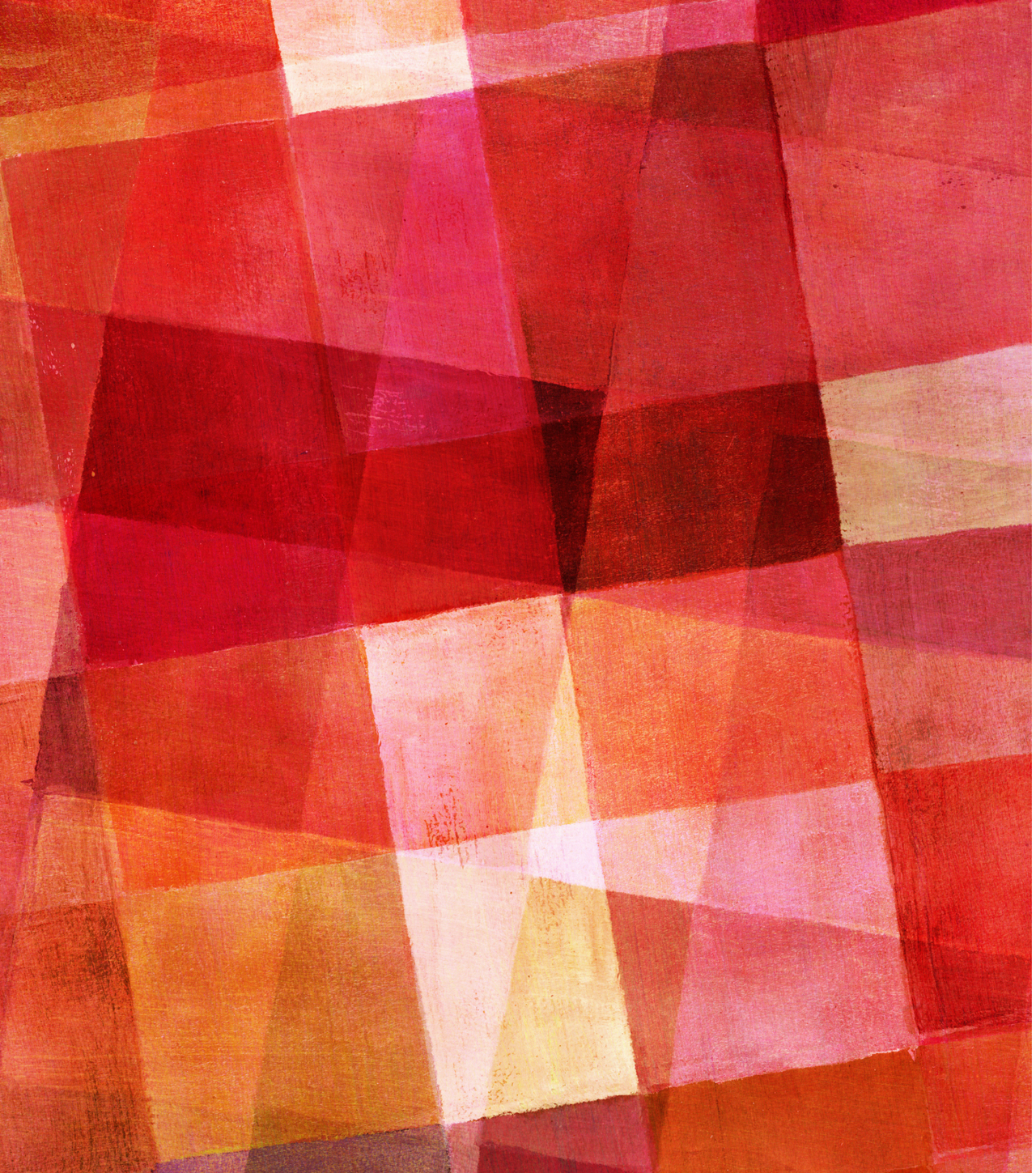
CINEL

Tecnologia e o Futuro num só Centr

SELECT + POWER

A função POWER retorna a potência de um número para outro número.

```
SELECT POWER(10,1) AS "Ten to the First",
    POWER(10,2) AS "Ten to the Second",
    POWER(10,3) AS "Ten to the Third";
```



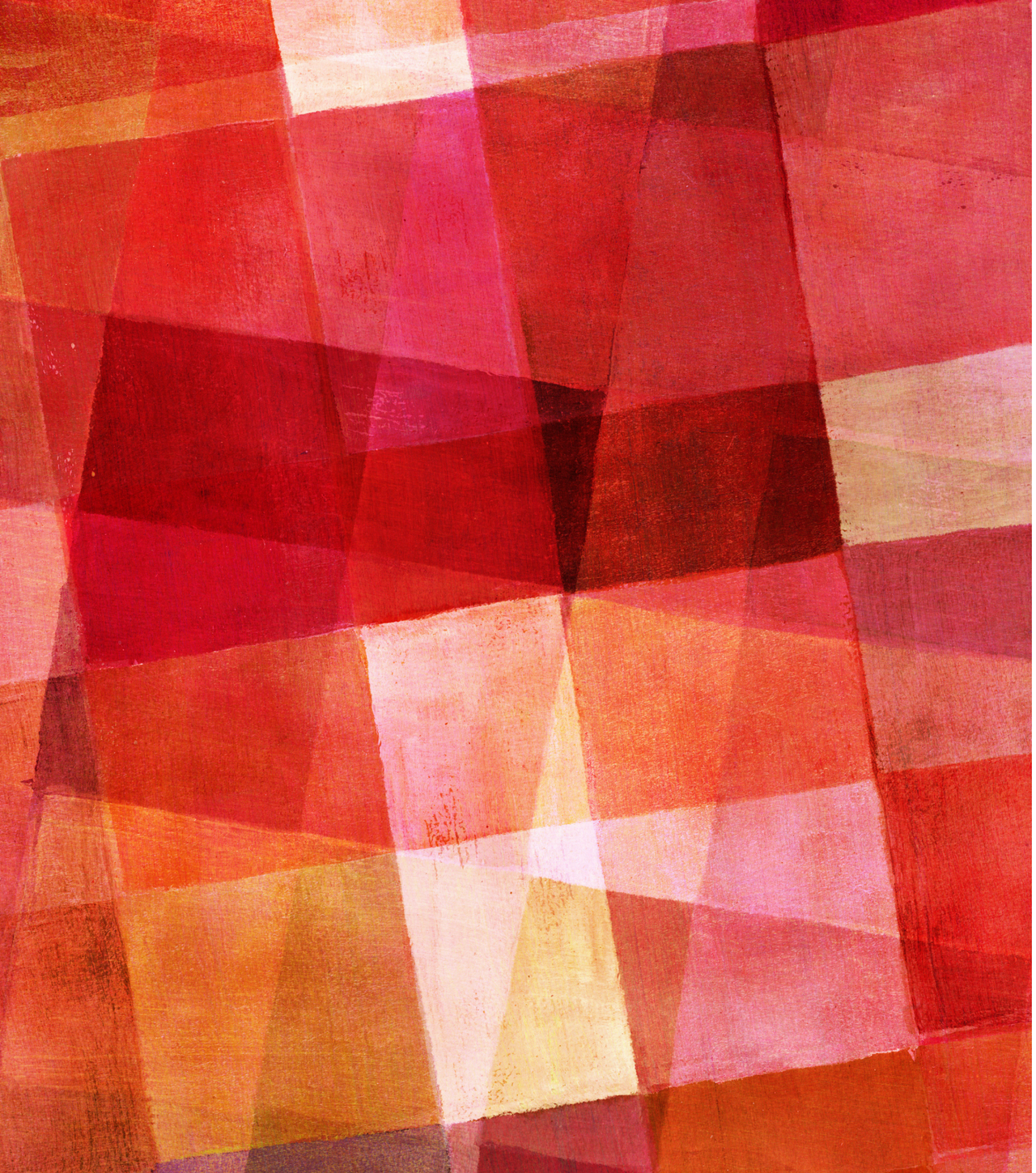
SQUARE/SQRT

SELECT + SQUARE/SQRT

A função SQUARE retorna o quadrado de um número ou o número multiplicado por ele mesmo.

A função SQRT retorna o oposto, a raiz quadrada de um número

```
SELECT SQUARE(10) AS "Square of 10",
    SQRT(10) AS "Square Root of 10",
    SQRT(SQUARE(10)) AS "The Square Root of the Square of 10";
```



ROUND

CINEL

Tecnologia e o Futuro num só Centr

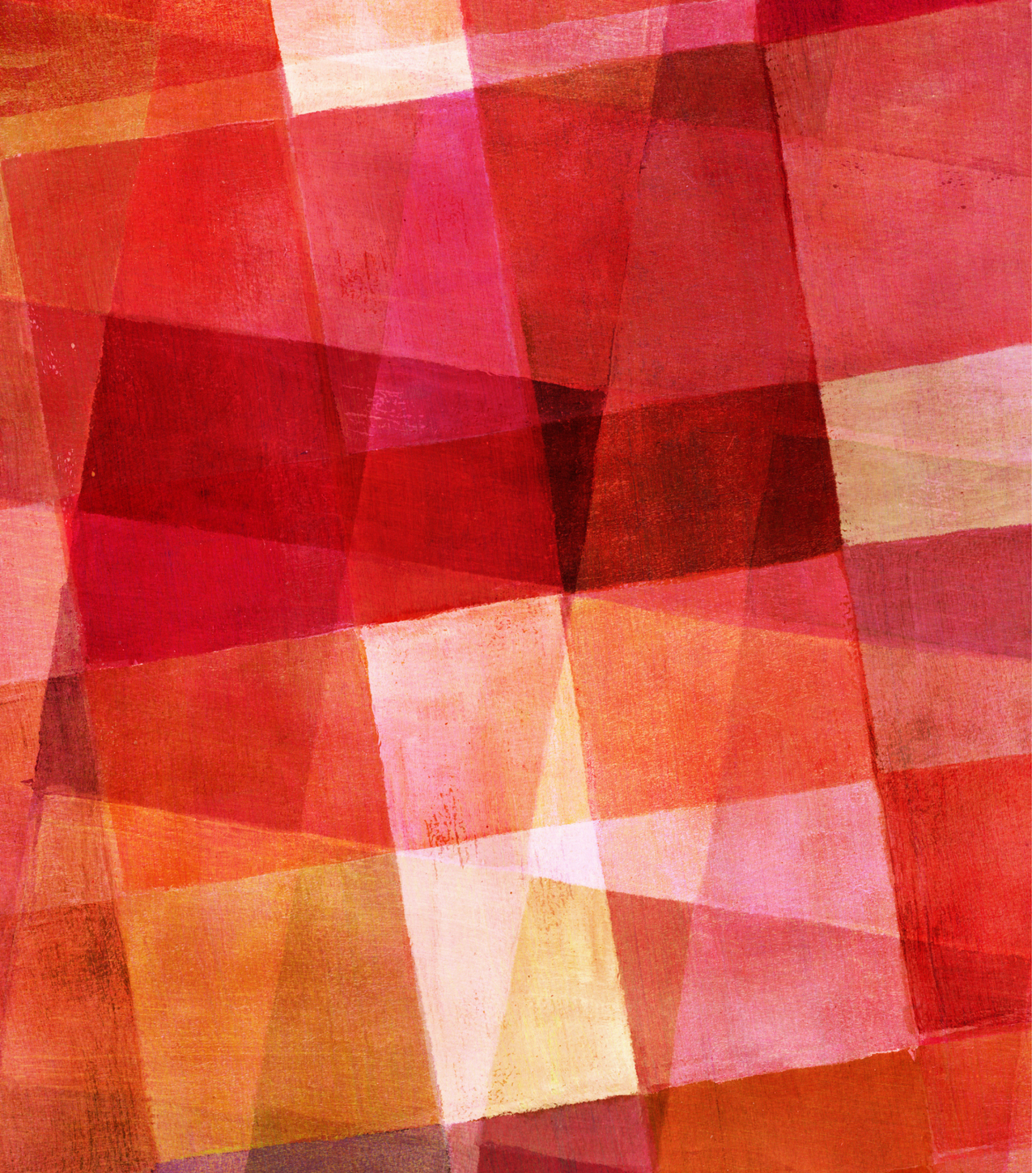
SELECT + ROUND

A função ROUND permite arredondar um número para uma determinada precisão.

A função ROUND é usada frequentemente para exibir apenas o número de casas decimais exigidas no relatório ou aplicativo. A função ROUND requer dois parâmetros, o número e o comprimento, que podem ser positivos ou negativos.

```
ROUND(<number>,<length>[,<function>])
```

```
SELECT ROUND(1234.1294,2) AS "2 places on the right",
    ROUND(1234.1294,-2) AS "2 places on the left",
    ROUND(1234.1294,2,1) AS "Truncate 2",
    ROUND(1234.1294,-2,1) AS "Truncate -2";
```



CASE

CINEL

Tecnologia e o Futuro num só Centr

SELECT + CASE

CINEL
Tecnologia e o Futuro num só Centr

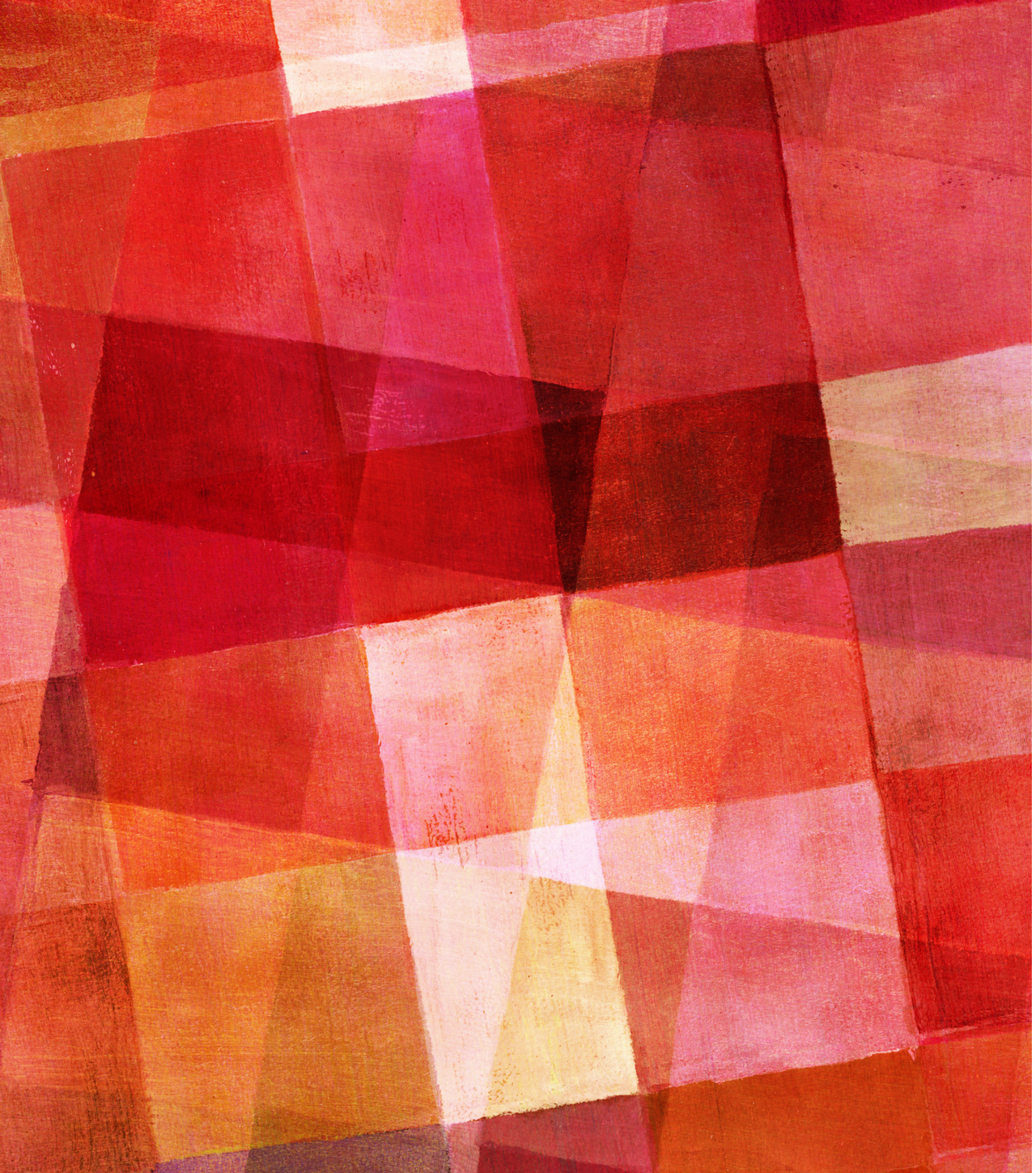
```
CASE <test expression>
    WHEN <comparison expression1> THEN <return value1>
    WHEN <comparison expression2> THEN <return value2>
    [ELSE <value3>] END
```

SELECT + CASE


```
SELECT Title,
    CASE Title
    WHEN 'Mr.' THEN 'Male'
    WHEN 'Ms.' THEN 'Female'
    WHEN 'Mrs.' THEN 'Female'
    WHEN 'Miss' THEN 'Female'
    ELSE 'Unknown' END AS Gender
FROM Person.Person
WHERE BusinessEntityID IN (1,5,6,357,358,11621,423);
```

SELECT + CASE SEARCH

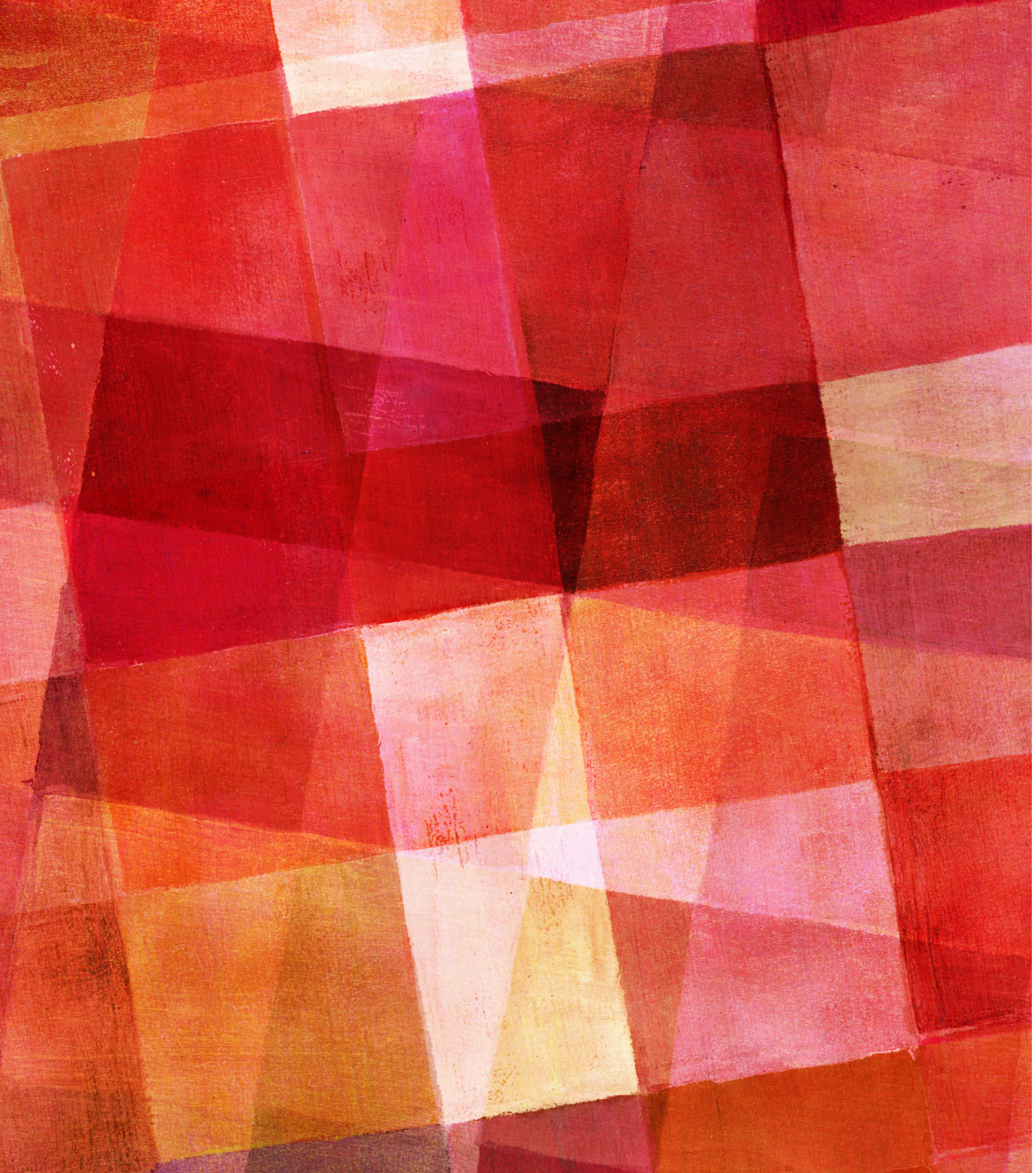

```
SELECT Title,
    CASE WHEN Title IN ('Ms.','Mrs.','Miss') THEN 'Female'
    WHEN Title = 'Mr.' THEN 'Male'
    ELSE 'Unknown' END AS Gender
FROM Person.Person
WHERE BusinessEntityID IN (1,5,6,357,358,11621,423);
```



GIF

SELECT + IIF

```
SELECT IIF (50 > 20, 'TRUE', 'FALSE') AS RESULT;


--2 IIF function with variables
DECLARE @a INT = 50
DECLARE @b INT = 25
SELECT IIF (@a > @b, 'TRUE', 'FALSE') AS RESULT;
```



ADMINISTRATIVE TOOLS

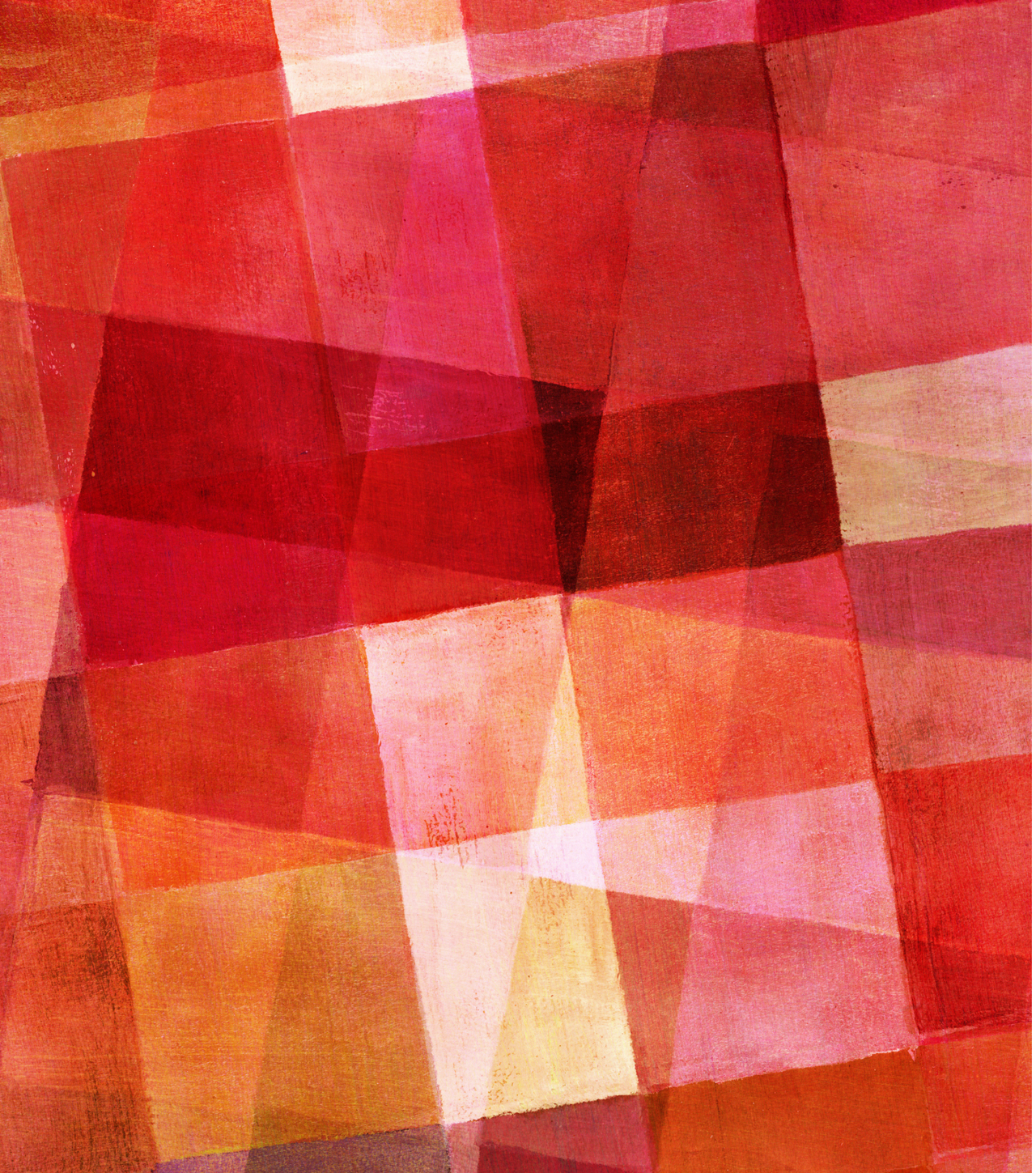
CINEL

Tecnologia e o Futuro num só Centr

SELECT + ADMINISTRATIVE TOOLS

Tecnologia e o Futuro num só Centr

```
SELECT
    @@VERSION 'Database Version',
    DB_ID() 'Database ID',
    DB_NAME() AS "Database Name",
    HOST_NAME() AS "Host Name",
    CURRENT_USER AS "Current User",
    SUSER_NAME() AS "Login",
    USER_NAME() AS "User Name",
    ORIGINAL_LOGIN() 'Original Login',
    USER 'User',
    SYSTEM_USER as 'System User',
    APP_NAME() AS "App Name";
```

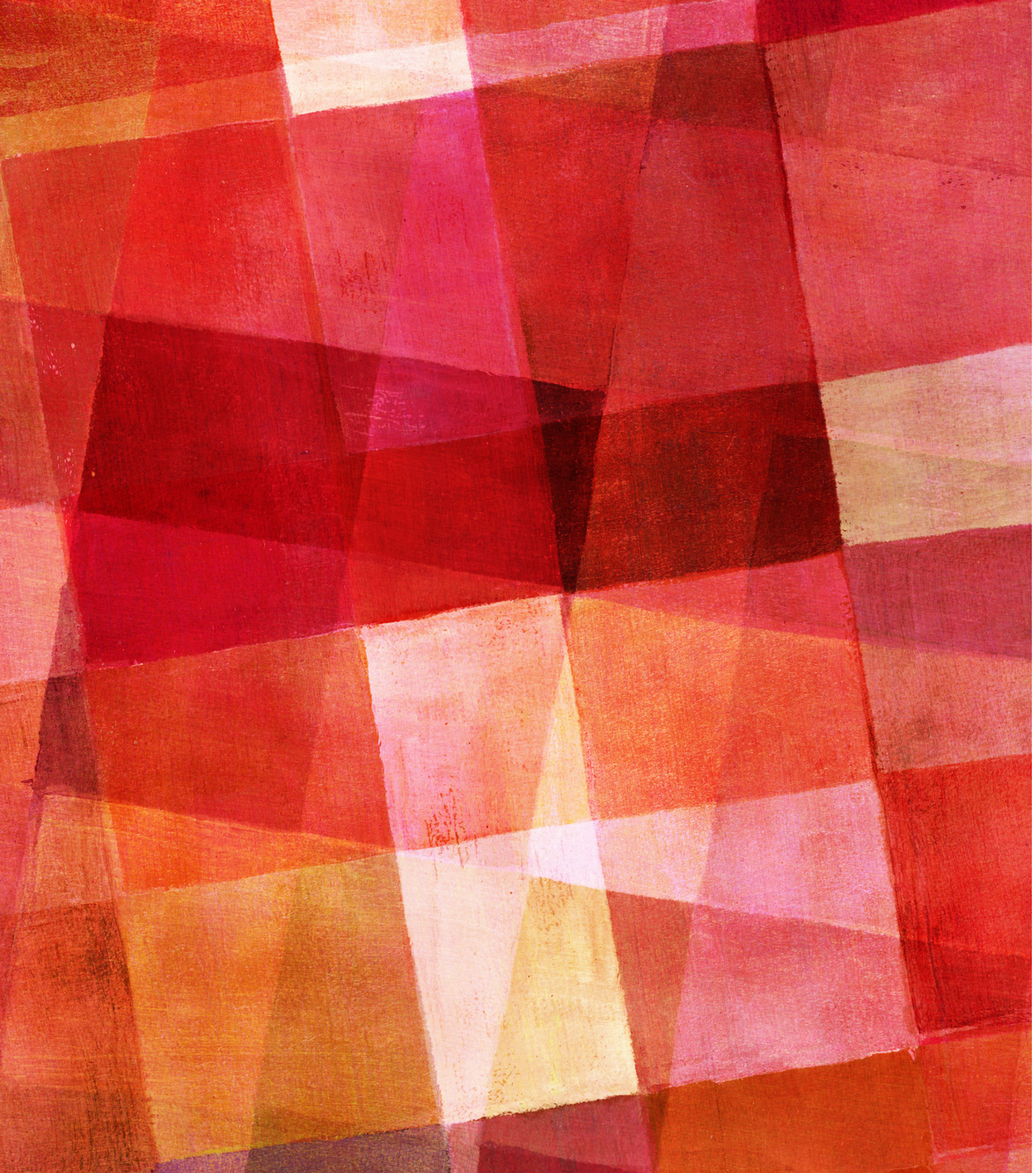


VARIABLES

SELECT + TOP QUERY COM VARIAVEIS

Tecnologia e o Futuro num só Centr

```
--1

DECLARE @Rows INT = 2;

SELECT TOP(@Rows) PERCENT CustomerID, OrderDate, SalesOrderID

FROM Sales.SalesOrderHeader

ORDER BY SalesOrderID;
```

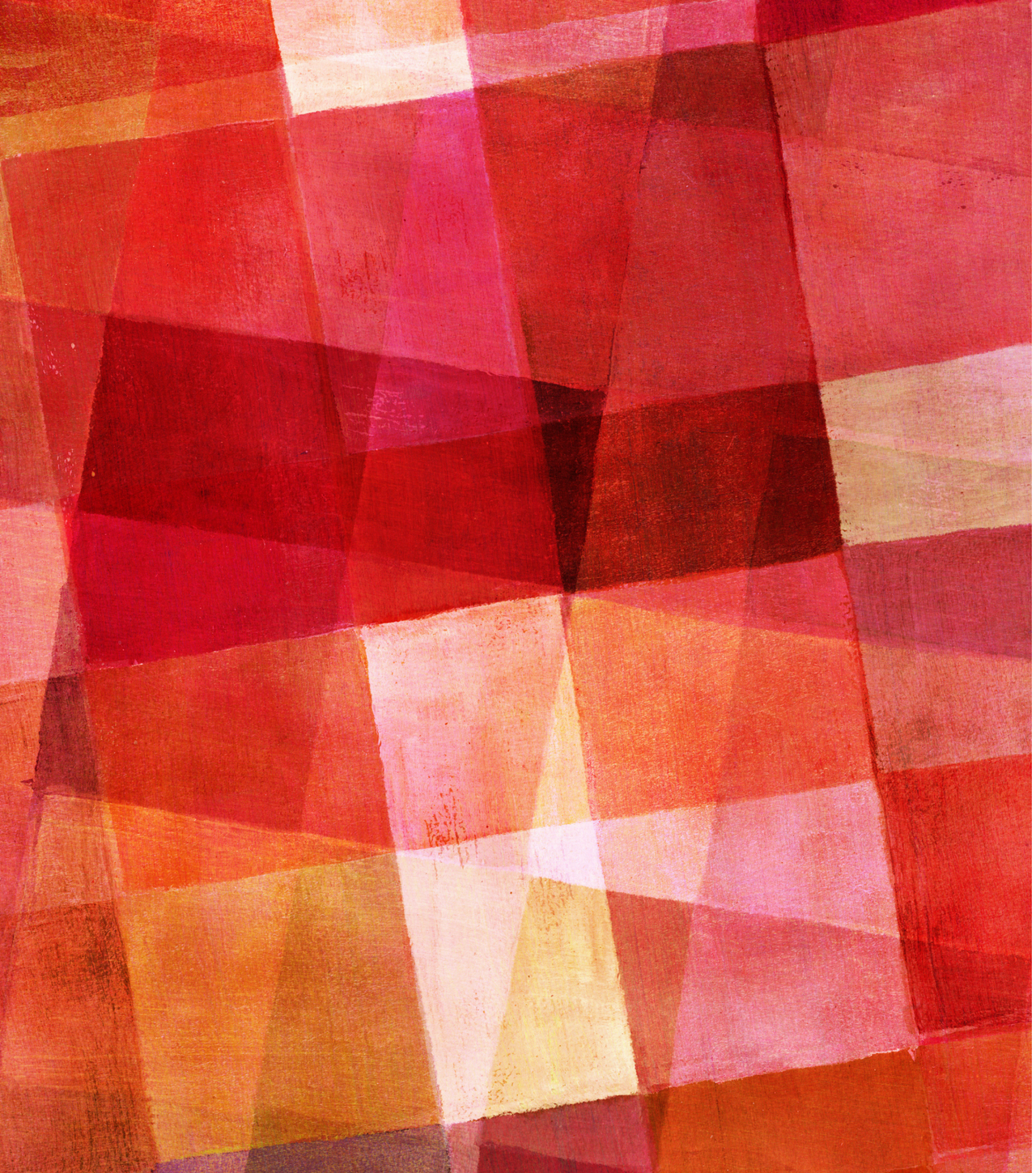


INDEX NON CLUSTERED

SELECT + CREATE INDEX

Tecnologia e o Futuro num só Centr

```
CREATE NONCLUSTERED INDEX [DEMO_SalesOrderHeader_OrderDate]
    ON [Sales].[SalesOrderHeader]
([OrderDate] ASC);

 IF  EXISTS (SELECT * FROM sys.indexes WHERE object_id =
     OBJECT_ID(N'[Sales].[SalesOrderHeader]')
     AND name = N'DEMO_SalesOrderHeader_OrderDate')
 DROP INDEX [DEMO_SalesOrderHeader_OrderDate]
 ON [Sales].[SalesOrderHeader] WITH ( ONLINE = OFF );
```



SUBQUERIES

SELECT + SUBQUERIES

```
SELECT cust_id
FROM Orders
WHERE order_num IN (SELECT order_num
                    FROM OrderItems
                    WHERE prod_id = 'RGAN01');
```